```
/** Singly linked list Ordered List.*/
public class OrSLinkedList {
  protected Node head;
  protected int size;
/********************/
  public OrSLinkedList()
{  head = null;
    size = 0;
  }
/********************/
  public int length()
{  Node c = head;
    int count = 0;
    while (c != null)
   {  count++;
      c = c.getNext();
    }
    return count;
  }
/********************/
  public boolean isEmpty() { return (head == null); }
/********************/
  public void delFirst()
{ if (isEmpty()) System.out.println("Nothing to Delete");
        else
                { head = head.getNext();
                size--;
                }
  }
/********************/
   public void delLast()
{       if (!isEmpty())
          {     Node c=head;
                Node prev=null;
                while (c.getNext() != null)
                    {   prev=c;
                        c = c.getNext();
                }
                if (prev == null)
                        head = null;
                else
                        prev.setNext(null);
                size--;
        }
        else
                System.out.println("Nothing to Delete");
  }

/********************/
  public int getFirst() {
    if (head == null)
      return 0;
    else
      return head.getElement();
  }
```

```
/*********************/
 public int getLast() {
   if (head == null){
        System.out.println("Nothing to return");
      return -1;
   }
   else {
      Node c = head;
      while (c.getNext() != null)
         c = c.getNext();
      return c.getElement();
   }
  }
/*********************/
  public String toString() {
   String result = "================\n";
   Node c = head;
   while (c != null) {
      result = result + c.getElement() + "\n";
      c = c.getNext();
   }
   return result;
  }
 /*********************/
  public boolean  find(int n)
  {
        Node c;
        for(c=head;c!=null;c=c.getNext())
        {
                if (c.getElement()==n)
                        return true;
        }
        return false;
  }
 /*********************/
 public void delIn(int n) {
        Node c=head;
        Node prev=null;
        while (c != null){
                if (c.getElement()==n)
                        break;
                else {
                        prev=c;
                        c = c.getNext();
                }
        }
        if (c != null){ //found
                if (prev == null)
                        head = head.getNext();
                else
                        prev.setNext(c.getNext());
                size--;
        }
        else
                System.out.println("Nothing to Delete");
```

```
 }


 /********************/
 public void insertO(int e) {
    Node n = new Node(e);
    Node c = head;
         Node prev = null;
         while (c != null)
          {      if (c.getElement()>e)
                        break;
                  else {
                        prev=c;
                        c = c.getNext();
                         }
         }
         if (prev = = null)
         {
                  n.setNext(head);
                  head=n;
         }
         else{
                  n.setNext(prev.getNext());
                  prev.setNext(n);
                  }
    size++;
 }

}
```

Main class

```
public class test
{
         public static void main (String args[])
         {
                  OrSLinkedList a = new OrSLinkedList();
                  a.insertO(1);
                  a.insertO(12);
                  a.insertO(7);
                  a.insertO(4);
                  System.out.println(a);
                  a.delIn(12); System.out.println(a);
                  while (!a.isEmpty())
                  {
                        System.out.println(a.getFirst());
                        a.delFirst();
                  }
                  System.out.println(a);
         }

}
```

رمزي القدومي